Predicting the Simulation Budget in Ranking and Selection Procedures

SIJIA MA and SHANE G. HENDERSON, Cornell University

The goal of ranking and selection (R&S) procedures is to identify the best among a finite set of alternative systems evaluated by stochastic simulation, providing a probability guarantee on the quality of the solution. To solve large-scale R&S problems, especially in parallel computing platforms where variable numbers of cores might be used, it is helpful to be able to predict the simulation budget, which is almost always the dominant portion of the running time of a given procedure for a given problem. Non-trivial issues arise due to the need to estimate the system configuration. We propose a set of methods for predicting the simulation budget. Numerical results compare our predictions for several leading R&S procedures.

CCS Concepts: • **Computing methodologies → Modeling and simulation**; *Simulation evaluation*; Simulation support systems;

Additional Key Words and Phrases: Ranking and selection, running-time prediction, stochastic optimization

ACM Reference format:
Sijia Ma and Shane G. Henderson. 2019. Predicting the Simulation Budget in Ranking and Selection Procedures. *ACM Trans. Model. Comput. Simul.* 29, 3, Article 14 (June 2019), 25 pages.
https://doi.org/10.1145/3323715

1 INTRODUCTION

Stochastic simulation is commonly used in the identification of the best system from a finite set of alternatives, where the performances of the systems are not explicitly available and can only be measured with error or noise. An example for such a problem is ambulance planning (Maxwell et al. 2013). The decision-makers of ambulance organizations need to decide the locations and schedules of the ambulances to minimize a quantile of the response times. For any candidate plan, its performance over a fixed time horizon is analytically intractable. Stochastic simulation can be used to approximately evaluate the objective function as part of an effort to solve this optimization problem. Such problems arise in, e.g., supply chain management (Jung et al. 2004) and transportation and public health (Banks 1998, part 4).

Ranking and selection problems arise when the number of feasible solutions is finite, and small enough that all systems can be simulated, at least to some degree. The main issue in these problems is how to allocate computational budget for simulation to the systems so that a statistically reliable

This research was supported, in part, by National Science Foundation Grants No. CMMI 1537394 and No. DMS-1839346 and Army Research Office Grant No. W911NF-17-1-0094.

Authors' addresses: S. Ma and S. G. Henderson, Cornell University, Operations Research and Information Engineering, 206 Rhodes Hall, Ithaca, NY, 14853, USA; emails: {sm2462, sgh9}@cornell.edu.

1049-3301/2019/06-ART14 $15.00
https://doi.org/10.1145/3323715

choice of the best system can be determined efficiently. Methods for such problems are called Ranking and Selection (R&S) procedures. A good R&S procedure achieves a balance between the total running time and the quality of the ultimate selection. The R&S problem is closely related to the multi-armed bandit (MAB) problem. An important difference is that, in MAB problems, one typically considers the cumulative reward collected by pulling each arm (i.e., obtaining a sample from each system) throughout the experiment, instead of focusing only on the quality of the final decision as in R&S (Peng et al. 2018). Even in pure-exploration instances of the MAB, that have the same overall objective as in R&S problems, different distributional assumptions are made about the underlying arms/systems. Algorithms developed for the bandit problem usually assume bounded or sub-Gaussian simulation outputs with a known bound on the variances, while R&S literature typically assumes Gaussian error, using batching to approximately ensure this when needed.

Many procedures have been proposed for dealing with R&S problems. The indifference-zone (IZ) approach, which assumes a known lower bound δ on the differences between the best system and all other systems, comprises a rich set of procedures, dating back to the seminal work of Bechhofer (1954) and Gupta (1956). Important example procedures include Dieker and Kim (2012), Dudewicz and Dalal (1975), Goldsman et al. (2002), Hong (2006), Kim and Dieker (2011), Kim and Nelson (2001), Kim and Nelson (2006), Rinott (1978), and Frazier (2014). These frequentist procedures guarantee a given pre-specified probability of correct selection (PCS), i.e., of returning the best system. There are also some recent IZ-free frequentist procedures (Fan et al. 2016) that deliver PCS without imposing an IZ restriction. Another form of statistical guarantee is the probability of good selection (PGS), to which we instead refer as probably approximately correct (PAC) selection, in line with the multi-armed bandit literature, that works for all configurations, and not just those that satisfy the IZ assumption; see, e.g., Nelson and Banerjee (2001) and Ni et al. (2014). Separately, a class of Bayesian approaches has been developed, including Chen et al. (2000), that allocates computational budget to maximize approximations of the posterior PCS. Other Bayesian procedures (Chick and Inoue 2001a, 2001b; He et al. 2007) allocate samples to minimize the expected opportunity cost, or to maximize a measure of the expected value of information (Chick et al. 2010).

Historically, due to limited computing capability, the interest in R&S problems primarily focuses on small-scale problems where the number of systems and the number of simulations of each system are relatively small. Nowadays, with improvements in computing power, the introduction of cloud computing and the development of efficient algorithms, much larger problems can be solved (Luo et al. 2015, Ni et al. 2017). However, new issues also emerge as the scale increases. Estimating the simulation budget, i.e., the number of replications of each system required to make a selection is key, because simulation budget is almost always the dominant contributor to running time, and the cost of running a task on the cloud depends on running time. Knowledge of the simulation budget also helps us decide the number of cores needed for the procedure to terminate in a reasonable time period in a parallel computing environment. For these reasons, a methodology for simulation-budget estimation for R&S procedures is important for large-scale problems.

The simulation budget of a procedure usually depends heavily on the configuration of the problem it is solving, i.e., the means and variances of these systems, which are unknown. Therefore a preliminary sampling stage is needed to get some knowledge about the structure of the problem. Non-trivial issues arise in this process. Since the sample size of the preliminary stage is usually very small, small differences in performances of systems are usually dominated by noise. With the naive way of estimating expectations by sample means, we tend to overestimate the differences between the best system and the others, and hence underestimate the running time. An extreme case is the slippage configuration in which all but one system have the same expectation and the other system is only a little better than the others. It is almost impossible to discern such a configuration with small sample sizes, since the sample means will vary. In this article, we propose a method to better deal with this "means-spreading" phenomenon, so that we have a better understanding about the true configuration of the problem. Our focus throughout the article is primarily on estimating the means of the systems, rather than the variances, because the means tend to have a more dramatic effect on running times. For example, to distinguish the difference between two normal populations having means μ_1 and μ_2, and variances σ_1^2 and σ_2^2, one can take n replications of each system until a statistically significant difference is seen in the sample means. The sample size n that is needed is on the order of $(\sigma_1^2 + \sigma_2^2)/(\mu_2 - \mu_1)^2$. Therefore, there is greater sensitivity to errors in predictions of means, rather than errors in predictions of variances, explaining our focus on means in this article.

Based on a limited sample size in the preliminary stage, it is extremely difficult to estimate the configuration of the problem and hence the simulation budget very accurately. Even if the system configuration is given, we still do not know the exact total number of samples due to the randomness of simulation output. However, our goal in estimating the simulation budget is to better allocate computing resources for solving the problem, so we only care about the order of magnitude of the simulation budget instead of its exact value. For example, we want to be able to predict whether the procedure will require 100 replications or 1,000 replications, rather than 100 replications or 110 replications. In the latter case, we are unlikely to alter the number of cores allocated.

In this article, we focus on estimating simulation budgets of procedures rather than the more desirable goal of estimating running times, partly because the two are so closely related. Moreover, there are complexities in estimating running times that make them difficult to analyze. For example, *synchronization* issues, the details of one's parallelization scheme and the implementation of the procedure all can heavily affect the wall-clock running time of procedures running in a parallel computing environment. We thus focus on estimating the simulation budget, taking the pragmatic, but not perfect, view that one can obtain an imperfect estimate of the running time by scaling simulation budgets by observed average running times per replication. This imperfect estimate does not account for the complexities mentioned above, but it can be viewed as an "approximate lower bound" on the running time, which suffices for our purpose.

There is an additional benefit of estimating the configuration of the problem. With a better understanding of the problem based on preliminary samples, we may be able to improve the efficiency of the R&S procedure. For example, for procedures with free parameters, e.g., Ni et al. (2017), taking advantage of the knowledge of the configuration helps us determine the values of parameters ($\bar{r}$ in Ni et al. (2017)) in such a way as to improve the procedure's efficiency. And for Bayesian procedures, the prior might be partially based on the estimated problem configuration.

The efficiency of a procedure might be impaired by the extra samples in the preliminary stage. Indeed, for small problems that can be solved in seconds, efficiency is probably not very important, and estimation of the running time might not be necessary. However, for large-scale problems the improvement of efficiency can be large even with the overhead brought by the extra samples. And there may be no need to discard the samples observed in the preliminary stage as long as we are not using the information provided by those samples for the procedure itself, e.g., to determine free parameters for the procedure. Therefore, we can just include those samples in the main body of the procedure and hence nothing is wasted.

Besides estimating the running time before the whole procedure starts, we can also update our estimate of the remaining time during the process to inform the user of progress through some kind of "progress bar." With more and more data observed, we expect that the estimate will be more and more accurate.

Our algorithm provides a point estimate of the problem configuration and the running time. One might desire more information, through confidence regions or belief distributions on the configuration, for example. Beyond repeatedly running our algorithm, we do not offer a practical method for obtaining such, more complex, information.

The article is organized as follows. Section 2 formally states the R&S problem and characterizes some properties of R&S procedures. Section 3 discusses two approaches for estimating the total number of samples for a given procedure and problem. The total number of samples depends on the estimation of ordered expectations, which is the goal of Section 4. Numerical experiments in Section 5 validate the quality of our estimation methods. Section 6 concludes.

2 PRELIMINARIES

Suppose we have k simulated systems, from which we can take samples by simulation to evaluate their performance. Let X_{ij} be the jth sample taken from System i, for $i = 1, \ldots, k$, $j = 1, 2, \ldots$. We assume $\mathbf{X}_i = (X_{ij} : j \geq 1)$ is an i.i.d. sequence of samples drawn from $\mathcal{N}(\mu_i, \sigma_i^2)$, a normal distribution with mean μ_i and variance σ_i^2, and moreover, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$ are independent. We denote $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_k^2)$ as the expectation vector and variance vector, which together are referred to as the *system configuration*. Our goal is to identify the system with largest expectation with a certain given level of confidence $1 - \alpha$.

There are two primary forms of frequentist probabilistic guarantees: the probability of correct selection (PCS) and probably approximately correct (PAC) selection. For a given confidence level $1 - \alpha$, the PCS guarantee means that the probability of choosing the best system (the system with the highest expectation) is no less than $1 - \alpha$, if the difference between the expectations of the best and others is greater than or equal to δ, the *indifference-zone parameter*. The PAC guarantee does not require such an assumption on the configuration of the problem. It means that the probability of selecting a system with expectation within δ of the best is no less than $1 - \alpha$.

In practice, simulation outputs may not be normally distributed, and then these probabilistic guarantees may not necessarily hold. A commonly used strategy is to draw a batch of samples at a time and take their mean as one output X_{ij}, so that the central limit theorem (CLT) ensures that the outputs are approximately normally distributed, and the guarantees might then hold approximately.

The sampling rule of a R&S procedure specifies how to allocate samples to each of the k systems, either deterministically or adaptively based on the samples that have already been observed. The R&S procedure terminates when the observed statistics are sufficient to deliver the pre-specified probabilistic guarantee. The sampling rule and stopping rule are designed to ensure the validity of the procedure, and together determine the efficiency of the R&S procedure, i.e., how many samples are needed to identify the best system.

Most efficient R&S procedures adopt an elimination strategy, i.e., identifying inferior systems to eliminate during the procedure as more samples are observed, so that those eliminated systems will not be considered as candidates and will no longer be sampled. At the end of the procedure, all but one system are eliminated, and the sole survivor is selected as the best.

3 ESTIMATING THE TOTAL NUMBER OF SAMPLES

R&S procedures essentially are composed of three tasks: (1) deciding the number of samples to draw from each system, (2) running simulations, and (3) screening (making elimination decisions), which includes computing and comparing statistics based on the simulation results. Usually, the computational cost for task (2) dominates, especially for large-scale problems, which is our focus in this article. Therefore, the essential question to ask in estimating the running time is how many samples are needed for the procedure to terminate and how long one replication takes.

The number of samples depends on the configuration of the given problem, i.e., the expectation vector $\boldsymbol{\mu}$ and the variance vector $\boldsymbol{\sigma}^2$, which are both unknown. Therefore, we use a preliminary stage to estimate the system configuration; see Section 4. For now, we just assume that we have already obtained an estimate of the system configuration, based on which we present two approaches for estimating the total number of samples: (1) a first-order estimate and (2) simulating the simulation process. The first approach simplifies the problem by ignoring the fluctuation in elimination time due to randomness in samples. The second approach simulates the whole R&S procedure, but instead of simulating the systems to get samples, it generates normally distributed samples directly from the estimated problem configuration.

In the preliminary stage, we estimate both the system configuration and the average simulation time $\bar{T}$, i.e., the amount of time spent taking one sample. Then the estimated total running time $\hat{T}_{\text{total}}$ is given by

$$\hat{T}_{\text{total}} = \hat{N}\bar{T},$$

where $\hat{N}$ is the estimated total number of samples. When the simulation time varies from system to system, we record $\bar{T}_i$'s separately for each System i, and then

$$\hat{T}_{\text{total}} = \sum_{i=1}^{k} \hat{N}_i \bar{T}_i,$$

where $\hat{N}_i$ is the estimated number of samples from System i.

In a parallel computing environment, other issues such as latency and synchronization also affect scalability and wall-clock time. Both highly depend on the algorithm itself, its implementation and the configuration of the problem. We ignore those issues, so that we get what might be viewed as an approximation of a lower bound on the running time when we convert our estimation to the parallel computing environment directly by dividing the total running time by the number of cores. In general, one might first estimate the total running time with our methodology, and second, decide the number of cores needed to finish the job in an acceptable time.

3.1 First-order Estimate

Under our assumptions the accumulated sums of the system outputs $(Y_i(n) : n \geq 0)$ with

$$Y_i(n) = \sum_{j=1}^{n} X_{ij}, i = 1, \ldots, k,$$

or the differences between pairs of such accumulated sums are random walks with normally distributed increments. R&S procedures explicitly (Hong 2006; Kim and Nelson 2001) or implicitly (Frazier 2014) define a continuation region, whereby a system is eliminated once the corresponding random walk hits the boundary $f(n, \theta)$ of the continuation region, where the parameter θ can involve statistics of the outputs, e.g., sample variances or accumulated sums of other systems. Hence, estimating N_i, the total sample size of System i, is equivalent to the problem of estimating a first-passage time of the corresponding random walk.

The expected first-passage time for simple cases (e.g., constant and linear boundaries) are well-understood, but other than that, closed-form formulae for few boundaries are known (Kahale 2008). For mathematical tractability, we simplify the problem to a first-order estimate, which means replacing the random walks $(Y_i(n), n \geq 0)$ by linear functions $(y_i(n) = n\mu_i, n \geq 0)$. In other words, we ignore the fluctuation of the random walks when estimating the first-passage time.

Why is this reasonable? The law of the iterated logarithm (Breiman 1992; Varadhan 2007) establishes that when $\sigma_i > 0$,

$$\limsup_{n\to\infty} \frac{|Y_i(n) - \mu_i n|}{\sigma_i \sqrt{n \log\log n}} = \sqrt{2}, \quad \text{a.s.}$$

Therefore for systems with large total sample size N_i, it is rare to see fluctuations go beyond the order of $\sqrt{n \log\log n}$ in magnitude, and the drift μ_i dominates. Therefore, relatively speaking, the performance of those systems will be close to their expected performance. When we focus on large-scale problems, most of the samples are contributed by those systems with large N_i, and so this simplification is reasonable. In addition, R&S procedures are designed to eliminate inferior systems based on their expectations irrespective of their variances, and so continuation regions are designed to be large enough to identify the average performance, irrespective of fluctuations, with high probability. Hence, it seems reasonable to neglect the randomness in determining when the random walk exits the continuation region. We will later see in numerical experiments that this seemingly rough simplification usually gives good results.

3.2 Simulation of the Simulation Process

Based on the estimated system configuration, we can simply run the whole R&S procedure for the original problem, replacing simulation by samples from a normal distribution with our estimated expectations and variances. In other words, we "simulate" the whole R&S procedure. Actually, this strategy has been used before to test R&S procedures, e.g., Kim and Nelson (2001) and Frazier (2014).

Usually R&S procedures spend a majority of their running time on simulations, which is skipped in this approach. Therefore, we expect it to be faster than the procedure itself, which is essential. The screening time of some R&S procedures is $O(k^2)$, for example, the original KN procedure (Kim and Nelson 2001) and (Hong 2006). For those procedures, this simulation approach may take a long time for large-scale problems, compared to those procedures with screening times that are linear (Frazier 2014; Hong et al. 2016) or approximately linear (Ni et al. 2017) in the number of systems. In these $O(k^2)$ procedures, we recommend using the first-order estimate approach if k is large.

4 ESTIMATION OF THE ORDERED EXPECTATIONS

4.1 Issues with Naive Estimation

As we see from the previous section, the estimation of the total number of samples N depends on the configuration $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ of the problem. The configuration is unknown, so we use a preliminary stage in which we sample each system n_0 times. For most R&S procedures that deal with unknown variances, there is such a stage for estimating variances. We can just make use of those samples taken for estimating variances for our preliminary stage so that no extra samples should be taken. However, if the estimate of running time will be used for determining the value of some free parameters in the R&S procedure, then usually those samples have to be dropped to ensure the statistical guarantee.

Recall that our goal is to answer questions like, how many samples are needed for the best system to eliminate the second best one. Therefore, instead of estimating the μ_i's, we only care about estimating the ordered expectations $\mu_{(i)}$'s without the need to know their identities. For σ_i^2's, we use sample variances as our estimators and pair them, in the natural way, with the ordered expectations by the order of the corresponding sample means. For example, if the sample mean $\bar{X}_i(n_0)$ of System i is the second largest, we pair its sample variance S_i^2 with the estimated second-largest ordered expectation $\hat{\mu}_{(2)}$. The same pairing strategy can be used for the average simulation time $\bar{T}_i$ as well.

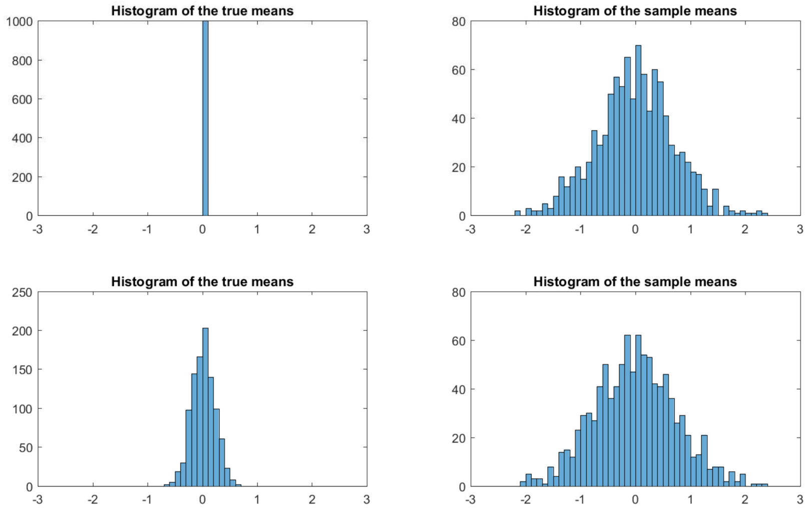

Fig. 1. Histograms of expectations and sample means under the following configurations: SC (row 1) and RPI (row 2). $k = 1{,}000$, $\delta = 0.1$, $\sigma_i^2 = 25$, $n_0 = 50$.

With n_0 samples from each of the systems, the most straightforward estimators of their expectations are the sample averages $\bar{X}_i(n_0)$, $i = 1, 2, \ldots, k$. Each sample average is an unbiased estimator of the expectation, but the estimated expectations tend to "spread out," so the ordered sample averages are not unbiased estimators of the ordered expectations. An extreme case is the slippage configuration (SC), in which $\mu_1 = \delta\ \mu_i = 0, \forall i \neq 1$, where $\sigma_i^2 = \sigma^2$ are equal. The sample means are independent and normally distributed with variance σ^2/n_0, hence the differences between their true means are exaggerated. We call this phenomenon "means-spreading." Histograms of expectations and sample means are given as illustration in Figure 1. The first row is for the SC, and the second row is for the random problem instances (RPI) configuration, where $\mu_i \sim \mathcal{N}(0, 4\delta^2)$ and are independent in i. As we can see, the sample means of both configurations in the right-hand panels are more spread out than their corresponding true expectations in the left-hand panels.

In general, for any configuration of the problem,

$$\mathbb{E} \max_i \bar{X}_i \geq \max_i \mathbb{E}\bar{X}_i = \max_i \mu_i, \text{ and}$$

$$\mathbb{E} \min_i \bar{X}_i \leq \min_i \mathbb{E}\bar{X}_i = \min_i \mu_i,$$

suggesting that means-spreading is ubiquitous. (We will later characterize the difference between $\mathbb{E} \max_i \bar{X}_i$ and $\max_i \mu_i$.) It is therefore reasonable to expect that we are being overly optimistic if we treat the sample means as true expectations, since the advantage of the best system over other systems is magnified, leading to an underestimation of $\hat{N}$. For configurations in which the expectations are close to each other, e.g., SC, the estimated total number of samples $\hat{N}$ can be tremendously smaller than what is needed, as we will see in numerical experiments.

The impact of means-spreading depends on how clustered the true expectations are relative to their true variances. If the expectations are very close to each other and the variances are large, then the sample means will be very spread-out. In general, for a fixed δ, the smaller the differences between the expectations and the larger the variances, the harder the problem, in the sense that it usually takes more samples to find the best system.

4.2 Regression Method for Estimating the Ordered Expectations

We have seen that recovering the ordered expectations solely from the sample means $\bar{X}_i(n_0)$ can perform badly. Fortunately, we have the full process: $(X_{ij}, i = 1, 2, \ldots, k, n = 1, 2, \ldots, n_0)$ and hence $(\bar{X}_i(n), i = 1, 2, \ldots, k, n = 1, 2, \ldots, n_0)$. Our goal is to understand the relation between ordered sample means and the ordered true expectations, especially those around $\mathbb{E} \max_i \bar{X}_i$ versus $\max_i \mu_i$.

First, consider the case where all the μ_i are the same. Assuming all X_i's are independent and normally distributed, the distribution function of $M = M(n) = \max_{i=1,\ldots,k} \bar{X}_i(n)$ is given by

$$\begin{aligned} F_M(x) &= P(M(n) \leq x) \\ &= \prod_{i=1}^{k} P(\bar{X}_i(n) \leq x) \\ &= \prod_{i=1}^{k} \Phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}}\right). \end{aligned}$$

The probability density function of M is then

$$\begin{aligned} f_M(x) &= \frac{d}{dx} \prod_{i=1}^{k} \Phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}}\right) \\ &= \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x-\mu}{\sigma_j/\sqrt{n}}\right)\right] \phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}}\right) \frac{1}{\sigma_i/\sqrt{n}}. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}M(n) &= \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x-\mu}{\sigma_j/\sqrt{n}}\right)\right] \phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}}\right) \frac{1}{\sigma_i/\sqrt{n}} dx \\ &= \mu + \frac{1}{\sqrt{n}} \int_{-\infty}^{\infty} y \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{y}{\sigma_j}\right)\right] \phi\left(\frac{y}{\sigma_i}\right) \frac{1}{\sigma_i} dy \quad \left(\text{let } y = \frac{x-\mu}{1/\sqrt{n}}\right) \\ &= \mu + \frac{C}{\sqrt{n}}, \end{aligned}$$

where

$$C = C(\boldsymbol{\sigma}) = \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x}{\sigma_j}\right)\right] \phi\left(\frac{x}{\sigma_i}\right) \frac{1}{\sigma_i} dx$$

is a constant that does not depend on n.

The value of C is usually unknown, since we do not know the variances σ_i. However, we do not need to calculate it explicitly, since we only care about the expectation μ. For each n, the corresponding largest sample mean $M(n)$ is known from the data, and so we can apply linear regression to estimate μ, using the model $M(n) = \mu + Cx(n)$ with $x(n) = n^{-1/2}$ to get an unbiased estimator of μ, which is the estimate of the intercept. An illustration of the regression method is given in Figure 2. In this example, the intercept of the fitted line, which is also the estimated $\hat{\mu}$, is 0.02, which is very close to its true value 0.

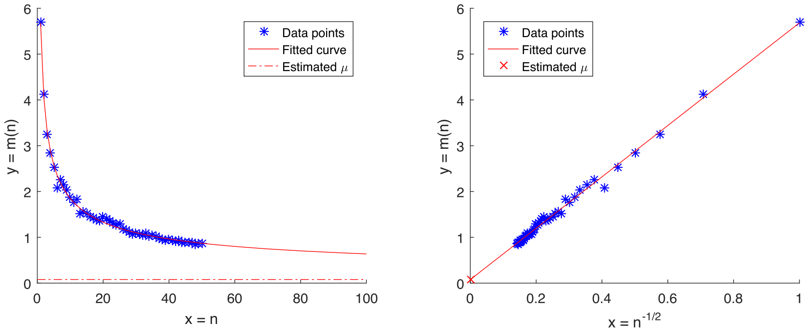

Fig. 2. Regression method to estimate μ when all $\mu_i = \mu = 0$ and $\sigma_i^2 = \sigma^2 = 4$ are equal. Here, $k = 1{,}000$ and $n_0 = 50$. The estimated $\hat{\mu} = 0.08$.

For general configurations, we similarly have

$$F_M(x) = \prod_{i=1}^{k} \Phi\left(\frac{x-\mu_i}{\sigma_i/\sqrt{n}}\right),$$

and

$$f_M(x) = \sum_{i=1}^{k}\left[\prod_{j\neq i}\Phi\left(\frac{x-\mu_j}{\sigma_j/\sqrt{n}}\right)\right]\phi\left(\frac{x-\mu_i}{\sigma_i/\sqrt{n}}\right)\frac{1}{\sigma_i/\sqrt{n}},$$

so that

$$\begin{aligned}
\mathbb{E}M(n) &= \int_{-\infty}^{\infty} x\sum_{i=1}^{k}\left[\prod_{j\neq i}\Phi\left(\frac{x-\mu_j}{\sigma_j/\sqrt{n}}\right)\right]\phi\left(\frac{x-\mu_i}{\sigma_i/\sqrt{n}}\right)\frac{1}{\sigma_i/\sqrt{n}}dx\\
&= \mu_{(k)} + \frac{1}{\sqrt{n}}\int_{-\infty}^{\infty} y\sum_{i=1}^{k}\left[\prod_{j\neq i}\Phi\left(\frac{y-\sqrt{n}(\mu_j-\mu_{(k)})}{\sigma_j}\right)\right]\\
&\quad\times\phi\left(\frac{y-\sqrt{n}(\mu_i-\mu_{(k)})}{\sigma_i}\right)\frac{1}{\sigma_i}dy \qquad \left(\text{let } y = \frac{x-\mu_{(k)}}{1/\sqrt{n}}\right)\\
&= \mu_{(k)} + \frac{D(n)}{\sqrt{n}},
\end{aligned}$$

where

$$\begin{aligned}
D(n) &= D(n,\boldsymbol{\mu},\boldsymbol{\sigma})\\
&= \int_{-\infty}^{\infty} x\sum_{i=1}^{k}\left[\prod_{j\neq i}\Phi\left(\frac{x-\sqrt{n}(\mu_j-\mu_{(k)})}{\sigma_j}\right)\right]\phi\left(\frac{x-\sqrt{n}(\mu_i-\mu_{(k)})}{\sigma_i}\right)\frac{1}{\sigma_i}dx.
\end{aligned}$$

In this case, the slope $D(n)$ is not constant. Moreover, its expression is complicated and mathematically intractable. What if we simply treat it as constant in n and apply linear regression as with the case of identical expectations? Denoting $\hat{\mu}_{(k)}$ as our estimator of $\mu_{(k)}$ by linear regression, the following theorem can help us understand the relation between $\hat{\mu}_{(k)}$ and $\mu_{(k)}$.

Theorem 1. *$D(n)$ is non-negative, non-increasing in n, and*

$$\lim_{n\to\infty} D(n,\boldsymbol{\mu},\boldsymbol{\sigma}) = C(\tilde{\boldsymbol{\sigma}}),$$

where the components of $\tilde{\boldsymbol{\sigma}}$ are those σ_i's whose corresponding μ_i's are equal to $\mu_{(k)}$.

Remark: The dimension of $\tilde{\boldsymbol{\sigma}}$ will usually be smaller than k. For example, if there are three systems with expectation equal to $\mu_{(k)}$, then the dimension of $\tilde{\boldsymbol{\sigma}}$ is 3, and implicit in our notation is the fact that we evaluate C with $k = 3$.

We use stochastic domination to prove this theorem. Recall that X is said to be *smaller than X' in the usual stochastic order* (denoted by $X \leq_{\text{st}} X'$) if

$$F'(x) \leq F(x) \quad \forall x \in \mathbb{R},$$

where F and F' are the cumulative distribution function of X and X', respectively.

Proof. (Theorem 1)

First, since

$$\mu_{(k)} + \frac{D(n)}{\sqrt{n}} = \mathbb{E} \max_{i=1,\ldots,k} \bar{X}_i(n) \geq \max_{i=1,\ldots,k} \mathbb{E}\bar{X}_i(n) = \mu_{(k)},$$

$D(n) \geq 0$. Second,

$$\begin{aligned} D(n) &= \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x - \sqrt{n}(\mu_j - \mu_{(k)})}{\sigma_j} \right) \right] \phi\left(\frac{x - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right) \frac{1}{\sigma_i} dx \\ &= \mathbb{E}\tilde{M}, \end{aligned}$$

where $\tilde{M} = \max_{i=1,\ldots,k} \tilde{X}_i$, $\tilde{X}_i \sim \mathcal{N}(\sqrt{n}(\mu_i - \mu_{(k)}), \sigma_i)$. The distribution function of $\tilde{M}$

$$F_{\tilde{M}(x)} = \prod_{i=1}^{k} \Phi\left(\frac{x - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right)$$

is non-decreasing in n for all x, since Φ is an increasing function and $\mu_i - \mu_{(k)} \leq 0, \forall i$, so the corresponding random variables are non-increasing in the usual stochastic order. Therefore, $D(n) = \mathbb{E}\tilde{M}$ is non-increasing. See Section (1.A.7) of Shaked and Shanthikumar (2007).

In addition, since

$$\lim_{n\to\infty} \Phi\left(\frac{x - \sqrt{n}(\mu_j - \mu_{(k)})}{\sigma_j} \right) = \begin{cases} \Phi\left(\frac{x}{\sigma_j}\right), & \mu_j = \mu_{(k)} \\ 1, & \mu_j < \mu_{(k)} \end{cases},$$

$$\lim_{n\to\infty} \phi\left(\frac{x - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right) = \begin{cases} \phi\left(\frac{x}{\sigma_i}\right), & \mu_i = \mu_{(k)} \\ 0, & \mu_j < \mu_{(k)} \end{cases},$$

$$\begin{aligned} \lim_{n\to\infty} D(n, \boldsymbol{\mu}, \boldsymbol{\sigma}) &= \lim_{n\to\infty} \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x - \sqrt{n}(\mu_j - \mu_{(k)})}{\sigma_j} \right) \right] \phi\left(\frac{x - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right) \frac{1}{\sigma_i} dx \\ &= \int_{-\infty}^{\infty} x \sum_{i:\mu_i = \mu_{(k)}} \left[\prod_{j \neq i, \mu_j = \mu_{(k)}} \Phi\left(\frac{x}{\sigma_j}\right) \right] \phi\left(\frac{x}{\sigma_i}\right) \frac{1}{\sigma_i} dx \\ &= C(\tilde{\boldsymbol{\sigma}}), \end{aligned}$$

where the components of $\tilde{\boldsymbol{\sigma}}$ are those σ_i's whose corresponding μ_i's are equal to $\mu_{(k)}$. The second equality can be justified by the Monotone Convergence Theorem, since $D(n, \boldsymbol{\mu}, \boldsymbol{\sigma})$ is monotonically decreasing in n. □

By Theorem 1, the slope $D(n)$ is non-negative, non-increasing with n and hence non-decreasing with $\frac{1}{n}$. Since our goal is to estimate the intercept of the linear function, in other words, the limit of $\mathbb{E}M(n)$ when $n \to \infty$, if we treat $D(n)$ as a constant, our estimator $\hat{\mu}_{(k)}$ underestimates the true highest expectation $\mu_{(k)}$. The larger n, the closer $D(n)$ to its limit, so using $M(n)$ with larger n in

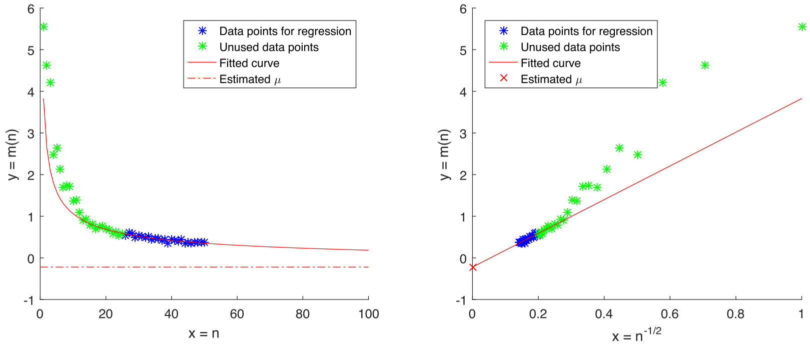

Fig. 3. Regression method to estimate $\mu_{(k)}$ when all $\sigma_i^2 = 4$ are equal and μ_i are independently and normally distributed with standard deviation equal to 0.2. We subtract $\max_i \mu_i$ from $\mu_i, i = 1, 2, \ldots, k$ to ensure that $\mu_{(k)} = 0$. Here, $k = 1{,}000$ and $n_0 = 50$. Only the last 25 data points are used for regression. The estimated $\hat{\mu}_{(k)} = -0.22$.

regression leads to lower bias. However, fewer data points brings higher variance, so there is a tradeoff between bias and variance over the number of data points used for regression. A practical rule of thumb that has had reasonable performance in our experiments is to use the second half of the data points, e.g., $(M(n), n = 26, 27, \ldots, 50)$ for $n_0 = 50$. An illustration is given in Figure 3, where the estimated $\hat{\mu}_{(k)} = -0.22$.

In our linear regression model, we use $M(n)$ in place of $\mathbb{E}M(n)$. However, using only one observation leads to high variance, especially for the case where n is small. If we have more observations of $M(n)$, then we can obtain a better estimate of $\mathbb{E}M(n)$ and hence make our estimation on $\mu_{(k)}$ more reliable. Limited to the sample size of the preliminary stage, we only have n_0 samples for each system. However, noticing that all the samples are generated independently and there is no specified order, we can use bootstrapping, e.g., we can shuffle the samples by randomly reordering them for each system to generate more data points $M(n)$'s for our regression, to make the estimation more stable. Denoting the number of shuffles by n_{shuffle}, for each $j = 1, 2, \ldots, n_{\text{shuffle}}$, we randomly reorder the samples for each system independently and compute the largest sample mean for the first n samples $M^{(j)}(n)$, $n = 1, 2, \ldots, n_0$. Define $\bar{M}(n) = \frac{1}{n_{\text{shuffle}}} \sum_{j=1}^{n_{\text{shuffle}}} M^{(j)}(n)$. Using $\bar{M}(n)$ instead of $M(n)$ gives a more accurate estimator of $\mathbb{E}M(n)$ and hence a more stable $\hat{\mu}_{(k)}$; see Figure 4.

The same idea applies to all of the ordered expectations $\mu_{(i)}$, $i = 1, 2, \ldots, k$. For each i, letting $M_i = M_i(n) = \bar{X}_{(i)}(n)$ be the ith smallest sample mean, its probability density function is given by

$$\sum_{r \in \mathcal{R}} \left[\prod_{j_1=1}^{i-1} \Phi\left(\frac{x - \mu_{r(j_1)}}{\sigma_{r(j_1)}/\sqrt{n}} \right) \right] \left[\prod_{j_2=i+1}^{k} \left(1 - \Phi\left(\frac{x - \mu_{r(j_2)}}{\sigma_{r(j_2)}/\sqrt{n}} \right) \right) \right] \phi\left(\frac{x - \mu_{r(i)}}{\sigma_{r(i)}/\sqrt{n}} \right) \frac{1}{\sigma_{r(i)}/\sqrt{n}},$$

where $\mathcal{R}$ is the set of permutations of $1, 2, \ldots, k$. Using the similar change of variable $y = \frac{x - \mu_i}{1/\sqrt{n}}$, we get

$$\mathbb{E}M_i(n) = \mu_{(i)} + \frac{D_i(n)}{\sqrt{n}},$$

Hence, we can estimate $\mu_{(i)}$ with linear regression as we did for $M = M_k$, although $D_i(n)$ is not necessarily monotone in n.

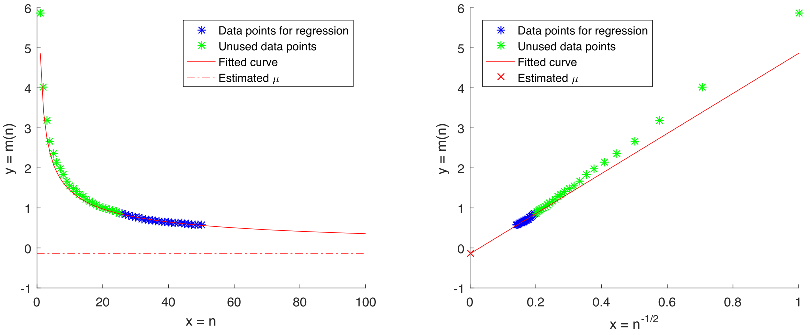

Fig. 4. Regression method to estimate $\mu_{(k)}$ when all $\sigma_i^2 = 4$ are equal and μ_i are normally distributed with standard deviation equal to 0.2, and $\mu_{(k)} = 0$. Here, $k = 1{,}000$ and $n_0 = 50$. Only the last 25 data points are used for regression. The estimated $\hat{\mu}_{(k)} = -0.14$.

Unfortunately, $\hat{\mu}_{(i)}$, the estimate of the ith order statistics, may not be monotonically increasing in i. Therefore, in practice, we further process the estimates $\hat{\mu}_{(i)}$ by taking $\hat{\mu}^{\mathrm{M}}_{(i)} = \max_{j=1,\ldots,i} \hat{\mu}_{(j)}$ to enforce monotonicity.

We are estimating the ordered means, so we want the variance of the differences between systems to be small. In practice, we recommend using common random numbers (CRN) for the initial stage to enhance the accuracy of the estimation. When CRN is employed, in the shuffling step, instead of shuffling each system's replications independently, we shuffle the replications across all systems together. More specifically, if $\mathbf{X}^j = \left(X_{ij} : i = 1, 2, \ldots, k\right)$, then we randomly select from $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^{n_0}$.

4.3 Convex Combination of Naive and Regression Estimators

As stated previously, $\hat{\mu}_{(k)}$ underestimates the true highest expectation $\mu_{(k)}$. Similarly, we can prove that $\hat{\mu}_{(1)}$ overestimates the true lowest expectation $\mu_{(1)}$. In general, intuitively and empirically, the estimated ordered expectations are condensed relative to the true configuration, which means that the estimated expectations will be closer to each other than the true expectations.

However, mean-spreading tells us that the ordered sample means are not condensed, but rather, spread out. That inspires us to combine these two consistent estimators to construct a less biased estimator. Here, we consider the linear combination (LC) estimator,

$$\hat{\boldsymbol{\nu}} = \rho \hat{\boldsymbol{\mu}}^{\mathrm{M}} + (1 - \rho)\boldsymbol{M},$$

where $\hat{\boldsymbol{\nu}}$ is our new estimator of the ordered expectations, $\hat{\boldsymbol{\mu}}^{\mathrm{M}} = (\hat{\mu}^{\mathrm{M}}_{(1)}, \ldots, \hat{\mu}^{\mathrm{M}}_{(k)})^T$ is the vector of the monotone regression estimators of the ordered expectations, $\boldsymbol{M} = (M_1(n_0), \ldots, M_k(n_0))^T$ is the vector of ordered sample means of the first n_0 samples, and ρ is a coefficient. To find an appropriate value of ρ for given vectors $\hat{\boldsymbol{\mu}}^{\mathrm{M}}$ and $\boldsymbol{M}$, we want to be able to measure how close $\hat{\boldsymbol{\nu}}$ is to the true ordered expectations, but these are unknown. However, we can take advantage of the ordered sample means in the sense that if two expectation vectors are close, then the ordered sample means they generate should also be close. For two ordered sample mean vectors $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$, we define the distance between them to be the L_2 norm of their difference:

$$\|\bar{\boldsymbol{x}} - \bar{\boldsymbol{y}}\| = \left[\sum_{i=1}^{k} (\bar{x}_{(i)} - \bar{y}_{(i)})^2\right]^{1/2}.$$

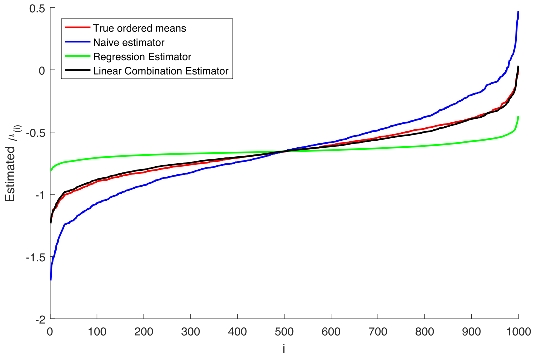

Fig. 5. Ordered expectations and the values of estimators. The red curve is the true ordered means, the blue curve is the naive estimator m_i, the green curve is the regression estimator $\hat{\mu}_{(i)}$ and the black curve is the linear combination estimator $\tilde{\mu}_{(i)}$. All $\sigma_i^2 = 4$ are equal and μ_i are independent and normally distributed with standard deviation equal to 0.2. Here, $k = 10{,}000$ and $n_0 = 50$. The coefficient $\rho = 0.48$.

We enumerate ρ from 0 to 1 in steps of length 0.02 to find the best coefficient. For each ρ, we first compute the corresponding $\tilde{\boldsymbol{\nu}}$, and generate n_{gen} sample means independently based on it, with the corresponding $\hat{\sigma^2}_i/\sqrt{n_0}$ as variance, then compute the average distance from those generated ordered sample means to the true ordered sample means $\boldsymbol{M}$. In our experiments, we set $n_{\text{gen}} = 100$. Finally, we select the ρ that minimizes this value, and take the corresponding $\tilde{\boldsymbol{\nu}}$ as our final estimator. As illustrated in Figure 5, comparing to the true ordered means, the naive estimator is more spread-out and the regression estimator is more condensed, while the linear combination of these two estimators with optimal coefficient $\rho = 0.4$ appears to be very close to the true values.

One might be tempted to simply set $\rho = 0.5$, which may be more efficient. However, the optimal ρ can be close to 0 when the means of the systems are spread-out, or close to 1 when the means of the systems are close to one another. In these cases, setting $\rho = 0.5$ does not yield accurate predictions. In addition, as we will show in numerical experiments, the time spent on predicting runtimes, including the time spent calculating a good choice of ρ, is negligible compared to the total running time of the R&S procedure.

Thus, we have a final estimator that appears to work much better than the original naive estimator. The whole algorithm is summarized below.

Estimation of the Ordered Expectations.

(1) Generate n_0 samples $X_{ij}, j = 1, 2, \ldots, n_0$ from each system $i = 1, 2, \ldots, k$. Compute $\bar{X}_i(n_0)$ and $\hat{\sigma^2}_i$ for each i. Let $M_i(n_0) = \bar{X}_{(i)}(n_0)$ and $r(i)$ be the corresponding index, i.e., $M_i(n_0) = \bar{X}_{r(i)}(n_0)$.

(2) For $j = 1, \ldots, n_{\text{shuffle}}$,
   - (a) For $i = 1, \ldots, k$,
     - (i) Shuffle $X_{ij}, j = 1, \ldots, n_0$ and get $X_{i\ell}^{(j)}, \ell = 1, \ldots, n_0$, where ℓ is the new index after shuffling.
     - (ii) Compute $\bar{X}_i^{(j)}(n) = \frac{1}{n}\sum_{\ell=1}^{n} X_{i\ell}^{(j)}$ for $n = 1, \ldots, n_0$.
   - (b) Compute the ordered sample means $M_i^{(j)}(n) = \bar{X}_{(i)}^{(j)}(n)$ for $i = 1, \ldots, k$, $n = 1, \ldots, n_0$.

(3) Compute $\bar{M}_i(n) = \frac{1}{n_{\text{shuffle}}}\sum_{j=1}^{n_{\text{shuffle}}} M_i^{(j)}(n)$ for $i = 1, \ldots, k$, $n = 1, \ldots, n_0$.

(4) Apply linear regression to calculate $\hat{\mu}_{(i)}$ using $\bar{M}_i(n), n = \lceil \frac{n_0}{2} \rceil + 1, \ldots, n_0$, for $i = 1, \ldots, k$.

(5) Enforce monotonicity by computing $\hat{\mu}_{(i)}^{\text{M}}$.

```
(6) For $\rho = 0, 0.02, \ldots, 1$,
    (a) Compute $\hat{\boldsymbol{\nu}} = \rho\hat{\boldsymbol{\mu}}^{\mathrm{M}} + (1-\rho)\boldsymbol{M}$.
    (b) For $j = 1, \ldots, n_{\text{gen}}$,
        (i) Generate $\bar{Y}_i \sim \mathcal{N}(\hat{\nu}_i, \widehat{\sigma^2}_{r(i)}/\sqrt{n_0})$ for $i = 1, \ldots, k$, independently.
        (ii) Compute $D_j(\rho)$, the $L_2$ distance between $(\bar{Y}_{(1)}, \bar{Y}_{(2)}, \ldots \bar{Y}_{(k)})^T$ and
             $\boldsymbol{M} = (M_1(n_0), M_2(n_0), \ldots, M_k(n_0))^T$.
    (c) Compute the average $L_2$ distance $D(\rho) = \frac{1}{n_{\text{gen}}}\sum_{j=1}^{n_{\text{gen}}} D_j(\rho)$.
(7) Find $\rho_{\text{opt}} = \operatorname{argmin}_\rho D(\rho)$ and let $\hat{\boldsymbol{\nu}} = \rho_{\text{opt}}\hat{\boldsymbol{\mu}}^{\mathrm{M}} + (1-\rho_{\text{opt}})\boldsymbol{M}$ be our estimate of the ordered
    true means. The estimate of the corresponding variances is $(\widehat{\sigma^2}_{r(1)}, \widehat{\sigma^2}_{r(2)}, \ldots, \widehat{\sigma^2}_{r(k)})^T$.
```

4.4 Statistical Properties of the Estimators

In this section, we give some theoretical support for the estimators. Specifically, we establish consistency and identify the rate of convergence of the naive estimator $M_i(n_0)$, the regression estimator $\hat{\mu}_{(i)}$ and hence the LC estimator $\hat{\nu}_i$. For simplicity, we analyze simpler forms of the estimators that exploit neither the shuffling of samples for generating more data points, nor the enforcing of monotonicity. In proving these results, we relax the normality assumption, since the results hold more generally. To simplify indexing notation, in this section, we assume, unbeknownst to our procedure, that $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$. We impose different subsets of the following conditions in the results to follow.

A0 Means are unique: $\mu_1 < \mu_2 < \cdots < \mu_k$.
A1 Sequences are iid with finite mean: $\mathbf{X}_i = (X_{ij} : j \geq 1)$ is an i.i.d. sequence with finite mean μ_i, for each $i = 1, \ldots, k$.
A2 Finite second moments: $\mathbb{E}(X_{i1}^2) < \infty$, so that the variance, σ_i^2, is finite, $i = 1, 2, \ldots, k$.
A3 Finite fourth moments: $\mathbb{E}(X_{i1}^4) < \infty$, $i = 1, 2, \ldots, k$.
A4 Mutual independence: $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$ are independent.

Proposition 1. *Fix $i \in \{1, 2, \ldots, k\}$. If conditions A0, A1 hold, then the naive estimator $M_i(n)$ is a strongly consistent estimator of μ_i, for each $i = 1, 2, \ldots, k$. If conditions A0–A2 hold, then we have the central limit theorem,*

$$\sqrt{n}(M_i(n) - \mu_i) \Rightarrow \sigma_i N(0, 1),$$

as $n \to \infty$. If conditions A0–A3 hold, then the mean squared error of $M_i(n)$ is $O(n^{-1})$.

Notice that we assume neither normality nor mutual independence; the result is more general, and covers these special cases. The assumption of finite fourth moments for the mean-squared error result is potentially stronger than needed, but permits a straightforward proof, and is standard; see, e.g., Glynn and Heidelberger (1990). Here, we assume the uniqueness of means, condition A0, but we relax that condition in Proposition 2 in the Appendix, leading to qualitatively similar results that are more complicated to establish. For example, the limiting distribution in the central limit theorem changes.

Proof. Let $\bar{X}_i(n) = n^{-1}\sum_{j=1}^n X_{ij}$ be the nth sample mean from System i. This does not necessarily equal $M_i(n)$, since $M_i(n)$ is the ith smallest sample mean. These two quantities agree on the event, $B_i(n)$ say, where $\bar{X}_i(n)$ is the ith smallest of the sample means $\bar{X}_1(n), \ldots, \bar{X}_k(n)$, i.e., it is ranked correctly, and may differ otherwise.

To establish consistency, the strong law of large numbers and the fact that k is finite ensure that $\max_{q=1}^k |\bar{X}_q(n) - \mu_q| \to 0$ as $n \to \infty$ a.s. Hence, since the μ_qs are unique, the event $B_i(n)$ occurs eventually a.s., i.e., there exists a (random) N such that for all $n \geq N$, $B_i(n)$ occurs a.s. Moreover, on the event $B_i(n)$, $M_i(n) = \bar{X}_i(n)$.

To establish the central limit theorem, let $\mathbb{I}\{B\}$ denote the indicator of the event B that equals 1 on the event and 0 otherwise. As noted above, $\mathbb{I}\{B_i(n)\} = 1$ for sufficiently large n (for $n \geq N$) a.s., so that the indicator of the complement, $\mathbb{I}\{B_i^c(n)\} = 0$ for $n \geq N$ a.s.. Define $E_i(n) = \sqrt{n}\,(M_i(n) - \mu_i)$. Then

$$E_i(n) = \sqrt{n}(\bar{X}_i(n) - \mu_i)\mathbb{I}\{B_i(n)\} + E_i(n)\mathbb{I}\{B_i^c(n)\},$$

and the result now follows from the ordinary central limit theorem and Slutsky's theorem.

To establish the mean-squared error result, we write

$$E_i(n) = \sqrt{n}(\bar{X}_i(n) - \mu_i) + \sqrt{n}(M_i(n) - \bar{X}_i(n))\mathbb{I}\{B_i^c(n)\}.$$

The inequality $(a+b)^2 \leq 2a^2 + 2b^2$ gives

$$\mathbb{E}(E_i(n)^2) \leq 2n\mathbb{E}(\bar{X}_i(n) - \mu_i)^2 + 2n\mathbb{E}\left[(M_i(n) - \bar{X}_i(n))^2\mathbb{I}\{B_i^c(n)\}\right]. \tag{1}$$

The first term in Equation (1) equals $2\sigma_i^2$. For the second term, the Cauchy-Schwarz inequality, together with the fact that $M_i(n)$ is equal to $X_J(n)$ for some (random) J gives

$$\begin{aligned}
&2n\mathbb{E}\left[\left(M_i(n) - \bar{X}_i(n)\right)^2 \mathbb{I}\{B_i^c(n)\}\right] \\
&\leq 2n\mathbb{E}\left[\left(M_i(n) - \bar{X}_i(n)\right)^4\right]^{1/2} \mathbb{P}(B_i^c(n))^{1/2} \\
&\leq 2n\left(\sum_{j\neq i} \mathbb{E}\left[\left(\bar{X}_j(n) - \bar{X}_i(n)\right)^4\right]^{1/2}\right)\left(\sum_{j>i}\mathbb{P}(\bar{X}_j(n) - \bar{X}_i(n) < 0) + \sum_{j<i}\mathbb{P}(\bar{X}_j(n) - \bar{X}_i(n) > 0)\right)^{1/2}.
\end{aligned}$$

We used the fact that $B_i^c(n) \subseteq \{\bigcup_{j>i}\{\bar{X}_j(n) - \bar{X}_i(n) < 0\} \cup \bigcup_{j<i}\{\bar{X}_j(n) - \bar{X}_i(n) > 0\}\}$. The term $\mathbb{E}[(\bar{X}_j(n) - \bar{X}_i(n))^4]$, by direct calculation, is $O(1)$. Chebyshev's inequality applied to the remaining terms yields the bound

$$\begin{aligned}
&2nO(1)\left(\sum_{j\neq i}\frac{\text{var}\,(\bar{X}_j(n) - \bar{X}_i(n))}{(\mu_j - \mu_i)^2}\right)^{1/2} \\
&= 2nO(1)\left(\sum_{j\neq i}\frac{\text{var}\,(X_j - X_i)}{n(\mu_j - \mu_i)^2}\right)^{1/2} \\
&= O(1),
\end{aligned}$$

concluding the proof. □

Turning now to the regression estimator, let $x_j = j^{-1/2}$ and $y_j = M_i(j)$ for $j \geq 1$. Assume that n is even, for notational simplicity. Define

$$\bar{x} = \frac{1}{n/2}\sum_{j=n/2+1}^{n} x_j,$$

so that $\bar{x}$ is a mean of the x_j values. Similarly, define $\bar{y}, \overline{xy}$ and $\overline{x^2}$, so that, for example,

$$\overline{xy} = \frac{1}{n/2}\sum_{j=n/2+1}^{n} x_j y_j.$$

The regression estimator is the y intercept of the estimated regression line, and can thus be written as

$$\hat{\mu}_i(n) = \frac{\overline{x^2}\bar{y} - \bar{x}\overline{xy}}{\overline{x^2} - \bar{x}^2}. \tag{2}$$

Theorem 2. *Fix $i \in \{1, 2, \ldots, k\}$. If condition A1 holds, then the regression estimator $\hat{\mu}_i(n)$ is a strongly consistent estimator of μ_i as $n \to \infty$. If conditions A0–A2 hold, then it satisfies the central limit theorem,*

$$\sqrt{n}(\mu_i(n) - \mu_i) \Rightarrow \eta N(0, 1),$$

where the variance constant η^2 is approximately $8.4144\sigma_i^2$. If conditions A1–A3 hold, then its mean-squared error is $O(n^{-1})$.

We assume uniqueness of the means to establish the central limit theorem, because the argument is involved; we conjecture that a version of the central limit theorem holds without that assumption.

Proof. To establish consistency, write

$$\hat{\mu}_i(n) = \bar{y} + \frac{\bar{x}^2(\overline{xy}/\bar{x} - \bar{y})}{\overline{x^2} - \bar{x}^2}. \tag{3}$$

From Proposition 2 in the Appendix, we know that under A1, $M_i(n) \to \mu_i$ as $n \to \infty$ a.s. It immediately follows that $\bar{y} \to \mu_i$ and $\overline{xy}/\bar{x} \to \mu_i$ as $n \to \infty$ a.s. For two real-valued sequences $(a_n : n \geq 1)$ and $(b_n : n \geq 1)$, we write $a_n \sim b_n$ if $a_n/b_n \to 1$ as $n \to \infty$. Approximating sums by integrals,

$$\begin{aligned}
\bar{x} &= \frac{2}{n} \sum_{j=n/2+1}^{n} j^{-1/2} \sim \frac{2}{n} \int_{n/2}^{n} x^{-1/2}\, dx = \frac{\beta_1}{\sqrt{n}}, \\
\overline{x^2} &= \frac{2}{n} \sum_{j=n/2+1}^{n} j^{-1} \sim \frac{\beta_2}{n}, \\
\overline{x^2} - \bar{x}^2 &\sim \frac{\beta_3}{n}, \text{ and} \\
\frac{\bar{x}^2}{\overline{x^2} - \bar{x}^2} &\sim \beta_1^2/\beta_3,
\end{aligned}$$

where the constants $\beta_1 = 4 - 2\sqrt{2}, \beta_2 = 2\ln(2)$, and $\beta_3 = \beta_2 - \beta_1^2$. These observations, together with the representation Equation (3), yield consistency.

We prove the mean-squared error result before the central limit theorem. Recall that we defined E_j so that $y_j = \mu_i + \frac{E_j}{\sqrt{j}}$. Using this definition in Equation (2),

$$\begin{aligned}
\hat{\mu}_i(n) &= \frac{\overline{x^2}\bar{y} - \bar{x}\overline{xy}}{\overline{x^2} - \bar{x}^2} \\
&= \frac{\overline{x^2}\left[\mu_i + \frac{2}{n}\sum_{j=n/2+1}^{n} j^{-1/2}E_j\right] - \bar{x}\left[\bar{x}\mu_i + \frac{2}{n}\sum_{j=n/2+1}^{n} j^{-1}E_j\right]}{\overline{x^2} - \bar{x}^2} \\
&= \mu_i + \sum_{j=n/2+1}^{n} \frac{2\overline{x^2}j^{-1/2} - 2\bar{x}j^{-1}}{n(\overline{x^2} - \bar{x}^2)} E_j \\
&= \mu_i + \sum_{j=n/2+1}^{n} c_{n,j}E_j, \qquad (4)
\end{aligned}$$

where $c_{n,j} = O(n^{-3/2})$ using the order-of-magnitude results above.

Thus, using the fact that $\mathbb{E}(E_n^2) = O(1)$ from Equation (6) in the proof of Proposition 2 in the Appendix, Equation (4) yields

$$\begin{aligned}\mathbb{E}[(\hat{\mu}_i(n) - \mu_i)^2] &= \mathbb{E}\left[\sum_{j=n/2+1}^{n} c_{n,j}E_j\right]^2 \\ &= O(1)\sum_{j=n/2+1}^{n}\sum_{m=n/2+1}^{n} c_{n,j}c_{n,m} \\ &= O(1)O(n^2)O(n^{-3}) \\ &= O(n^{-1}),\end{aligned}$$

giving the mean-squared error result.

Turning to the central limit theorem, by consistency and the fact that the means are assumed unique, there exists (random) N such that for all $j \geq N, M_i(j) = \bar{X}_i(j)$. Thus, $\mathbb{I}\{n/2 \geq N\}$ converges to 1 almost surely, and Equation (4) gives

$$\sqrt{n}(\hat{\mu}_i(n) - \mu_i) = \mathbb{I}\{n/2 \geq N\}\sqrt{n}\sum_{j=n/2+1}^{n} c_{n,j}E_j + \mathbb{I}\{n/2 < N\}\sqrt{n}(\hat{\mu}_i(n) - \mu_i).$$

The second term in this expression is eventually equal to 0 and the indicator in the first expression is eventually equal to 1. It therefore suffices to focus attention on the summation in the first expression. Moreover, on the event $\{n/2 \geq N\}$, $E_j = \sqrt{j}(\bar{X}_i(j) - \mu_i)$ for $j \geq n/2$, and then (on that event),

$$\begin{aligned}\sqrt{n}\sum_{j=n/2+1}^{n} c_{n,j}E_j &= \sum_{j=n/2+1}^{n} c_{n,j}\sqrt{nj}(\bar{X}_i(j) - \mu_i) \\ &= \sum_{j=n/2+1}^{n} c_{n,j}\sqrt{n/j}\sum_{m=1}^{j}(X_i(m) - \mu_i) \\ &= \sum_{m=1}^{n}\left(\sum_{j=\max(n/2+1,m)}^{n} c_{n,j}\sqrt{n/j}\right)(X_i(m) - \mu_i) \\ &= \sum_{m=1}^{n} a_{n,m}(X_i(m) - \mu_i),\end{aligned}$$

for appropriately defined constants $a_{n,m}$, where we changed the order of summation in the second-to-last step. We can now apply the Lindeberg-Feller Central Limit Theorem; see, e.g., Theorem 27.2 of Billingsley (1995). Verifying the Lindeberg condition is straightforward; it follows exactly the same lines as the discussion of (27.9) in that text. The central limit theorem we desire, therefore, follows using the magnitude of the constants $a_{n,m}$ that we determine next, and has variance given by

$$\sigma_i^2 \lim_{n\to\infty}\sum_{m=1}^{n} a_{n,m}^2. \tag{5}$$

It remains to obtain the expression for the variance. To that end,

$$\begin{aligned} a_{n,m} &= \sum_{j=\max(n/2+1,m)}^{n} c_{n,j}\sqrt{n/j} \\ &= 2\sqrt{n} \sum_{j=\max(n/2+1,m)}^{n} \frac{\overline{x^2}j^{-1} - \bar{x}j^{-3/2}}{n(\overline{x^2} - \bar{x}^2)} \\ &\sim \frac{2\sqrt{n}}{\beta_3} \sum_{j=\max(n/2+1,m)}^{n} (\beta_2 n^{-1} j^{-1} - \beta_1 n^{-1/2} j^{-3/2}). \end{aligned}$$

Using an integral to asymptotically approximate the sum shows that for $m \leq n/2$,

$$a_{n,m} \sim \frac{2\beta_2 \ln(2) - 4\beta_1(\sqrt{2} - 1)}{\beta_3\sqrt{n}} = \frac{\beta_4}{\sqrt{n}},$$

and for $m > n/2$,

$$a_{n,m} \sim \frac{2}{\beta_3\sqrt{n}} \left(\beta_2 \ln(n/m) - 2\beta_1 \left(\sqrt{n/m} - 1\right)\right).$$

We now use an integral once more to obtain the asymptotic form of Equation (5). This step involves considerable but tedious algebra. A symbolic manipulator yields the expression

$$\begin{aligned} \sigma_i^2[&-512\sqrt{2} + 2208\sqrt{2}\ln^2(2) - 904\sqrt{2}\ln^3(2) + 80\sqrt{2}\ln^4(2) + 2688\ln^2(2) - 1392\ln^3(2) + 84\ln^4(2) \\ &- 4\ln^5(2) - \ln(2)(384\sqrt{2} - 256) - 1024]/(\ln^2(2) - 24\ln(2) + 16)^2, \end{aligned}$$

which evaluates to $8.4144\sigma_i^2$. □

The observation in Theorem 2 that the asymptotic variance of the regression estimator (that appears in the central limit theorem) is larger than that of the naive estimator should not be too surprising. Indeed, the regression estimator is designed to combat the bias in the naive estimator, and bias-reduced estimators usually entail increased variance. Moreover, the regression estimator is essentially a weighted average of the samples, and with non-uniform weights one would expect an increase in variance. The fact that the asymptotic variance only involves σ_i^2 and not the variance of the other systems is natural; the asymptotic analysis depends on the fact that asymptotically the systems are correctly ordered by their sample means, and on that event the regression estimator is based only on samples from the ith system.

The LC estimator is a convex combination of the naive and linear regression estimators, where the weight ρ is random and is obtained from the same data used to construct the estimators. Accordingly, the results for consistency and mean-squared error are easily established. If the estimate of ρ converges a.s. as the sample size goes to infinity, as would be the case if it is fixed at the outset for example, then a central limit theorem for the LC estimator can also be established (Peng et al. 2017), but we do not provide that result, since it is a straightforward extension of the results we have established.

Corollary 1. *Under condition A1, the LC estimator $\hat{v}_i$ is a strongly consistent estimator of μ_i. Under conditions A1–A3, its mean-squared error is $O(n^{-1})$.*

5 NUMERICAL EXPERIMENTS

In this section, we demonstrate the performance of our estimates of the simulation budget as measured through the total number of samples. We use three efficient R&S procedures: the KN procedure (Kim and Nelson 2001), the Bayes-inspired indifference zone (BIZ) procedure (Frazier

Table 1. Problem Configurations Used in Numerical Experiments

Name	Expectation	Variances
Slippage Configuration (SC)	$\mu = [\delta, 0, \ldots, 0]$	$\sigma_i^2 = \sigma^2 = 4$
SC, increasing variances (SC-INC)	$\mu = [\delta, 0, \ldots, 0]$	$\sigma_i^2 = (1 + \frac{2i}{k})^2$
SC, decreasing variances (SC-DEC)	$\mu = [\delta, 0, \ldots, 0]$	$\sigma_i^2 = (3 - \frac{2i-2}{k})^2$
Monotone decreasing means (MDM)	$\mu_i = -\delta(i-1)$	$\sigma_i^2 = \sigma^2 = 4$
MDM, increasing variances (MDM-INC)	$\mu_i = -\delta(i-1)$	$\sigma_i^2 = (1 + \frac{2i}{k})^2$
MDM, decreasing variances (MDM-DEC)	$\mu_i = -\delta(i-1)$	$\sigma_i^2 = (3 - \frac{2i-2}{k})^2$
Random problem instances 1 (RPI1),	$\mu_i \sim \mathcal{N}(0, 4\delta^2)$, i.i.d.	$\sigma_i^2 = \sigma^2 = 4$
Random problem instances 2 (RPI2),	$\mu_i \sim \mathcal{N}(0, 25\delta^2)$, i.i.d.	$\sigma_i^2 = \sigma^2 = 4$
RPI, heterogeneous variances (RPI-HET),	$\mu_i \sim \mathcal{N}(0, 25\delta^2)$, i.i.d.	$\sigma_i^2 = (1 + \frac{2i}{k})^2$

2014) and the Good Selection Procedure (GSP) (Ni et al. 2017). The KN procedure is highly influential and might be considered the state-of-the-art for indifference zone R&S (Branke et al. 2007). It compares all pairs of systems and eliminates inferior ones when certain random walks leave a triangular continuation region, until one system remains. The BIZ procedure has a tight worst-case preference-zone PCS bound for the known-variance case, and is empirically more efficient than KN on a variety of problems. The GSP is designed for a parallel computing environment, and it provides a PAC guarantee. Since we are only using the samples taken in the preliminary stage for simulation budget prediction, those samples are reused as initial samples in the R&S procedures. All the source code for our experiments is available in the open-source repository Ma (2016).

5.1 Comparison of Naive Estimation and Linear Combination Estimation

We demonstrate the difference in performance of the naive estimator and LC estimator on nine standard test configurations: SC, SC-INC, SC-DEC, MDM, MDM-INC, MDM-DEC, RPI1, RPI2, RPI1-HET, with three problem sizes: $k = 100, 500, 2{,}000$. These test configurations (and their acronyms) are described in Table 1. For all configurations and problem sizes, the indifference zone constant $\delta = 0.1$ and the confidence level $1 - \alpha = 0.95$. We compare and illustrate the accuracy of the estimators by the ratios of the estimator and the true value of the total number of samples N. The results do not show significant difference between the three procedures, BIZ, KN and GSP, so we only visualize the results for BIZ through histograms of the ratios under different configurations with different k, as shown in Figures 6, 7, and 8. The results in the histograms come from the *simulation of the simulation process* approach. The average ratio for all procedures and both *first-order estimate* and *simulation of the simulation process* approaches are given in Tables 5 and 6 in the Appendix.

For the slippage configurations in Figure 6, the LC estimator performs consistently much better than the naive estimator. We see that we usually underestimate the number of samples, perhaps because the slippage configuration is difficult to recover. However, there is still a huge difference between the performances of the naive estimator and linear combination estimator. The LC estimator can achieve around 70% accuracy (the ratio of the estimator to the true value), and so we expect that it can get the correct order of magnitude, which is enough for determining the number of cores or how much computing resource is needed in practice. In comparison, the value of the naive estimator is usually around 10% to 20% of the true value, and that can result in an incorrect order of magnitude.

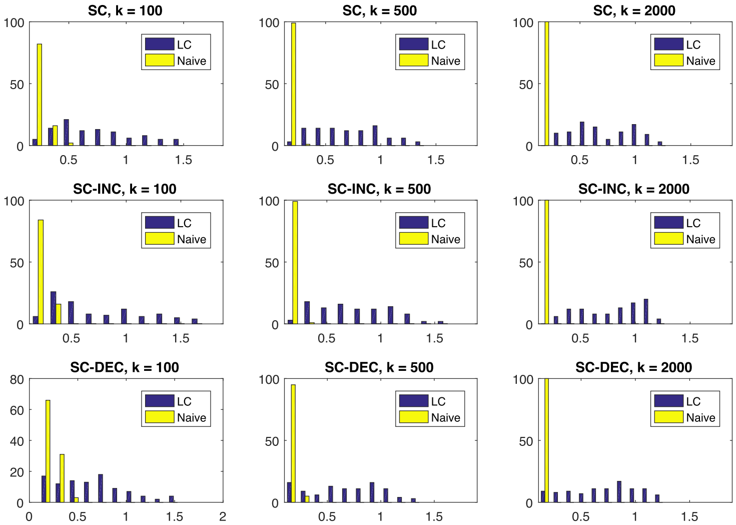

Fig. 6. Histogram of ratios between estimation and true value of N. Test on BIZ under the following configurations: SC (row 1), SC-INC (row 2), and SC-DEC (row 3); with $k = 100$ (column 1), 500 (column 2), and 2,000 (column 3).

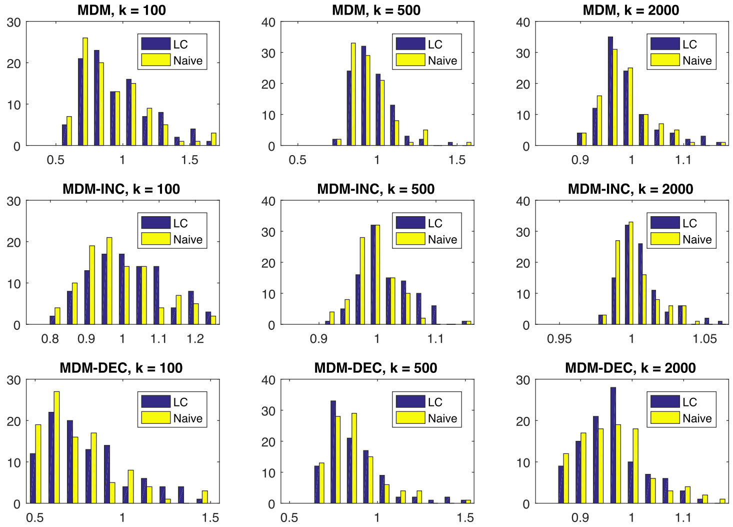

Fig. 7. Histogram of ratios between estimation and true value of N. Test on BIZ under the following configurations: MDM (row 1), MDM-INC (row 2), and MDM-DEC (row 3); with $k = 100$ (column 1), 500 (column 2), and 2,000 (column 3).

For monotone decreasing means configurations in Figure 7, both estimators perform well. The reason is that the means are already very clearly separated, so the “mean-spreading” phenomenon is not severe in this case. In practice, we expect that both estimators should have good performance for configurations with spread-out means.

For the random problem instances in Figure 8, the LC estimator performs well while the naive estimator sometimes diverges from the truth. In those random instances when there are many

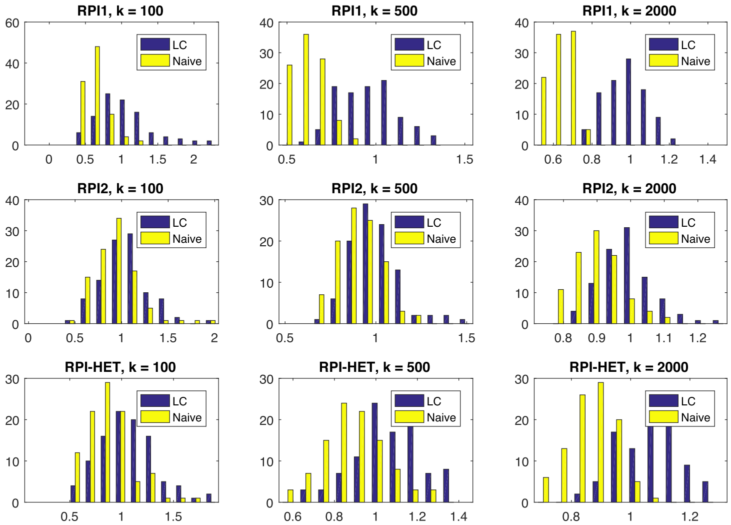

Fig. 8. Histogram of ratios between estimation and true value of N. Test on BIZ under the following configurations: RPI1 (row 1), RPI2 (row 2), and RPI-HET (row 3); with $k = 100$ (column 1), 500 (column 2), and 2,000 (column 3).

good systems not too far away from the best, the LC estimator is still able to recover the original configuration, while the naive estimator might be confused. The ratios between the LC estimator and the true values are usually between 0.8 and 1.2, especially for large k. In comparison, the naive estimator tends to underestimate.

In practice, we rarely see a real-world problem that is as difficult as the slippage configuration. Usually the means are more spread out than the slippage configuration but more condensed than the MDM configuration. In general, the LC estimator should be able to accurately reflect the true configuration without being too "optimistic" or "pessimistic" about the running time.

The LC estimator performs better when k is large for condensed configurations (SC family and RPI1), since with more information provided by more systems, it is able to more accurately estimate the configuration. This is desirable, since we mainly focus on large-scale problems.

As shown in Table 6, the estimations obtained with the first-order approach are similar to those obtained with the simulation approach and also have good performance.

5.2 An Illustrative Problem

In this section, we test our estimation of the total number of samples on the "throughput-maximization" problem taken from SimOpt.org (Henderson and Pasupathy 2014). In this problem, we want to solve

$$\begin{aligned}
&\max_x \mathbb{E}\left[g(x;\xi)\right], \\
\text{s.t. } & r_1 + r_2 + r_3 = R, \\
& b_1 + b_2 = B, \\
& x = (r_1, r_2, r_3, r_4, r_5) \in \{1, 2, \ldots\}^5,
\end{aligned}$$

where the function $g(x;\xi)$ represents the random throughput of a three-station flow line with finite buffer storage in front of Stations 2 and 3, denoted by b_2 and b_3, respectively, and an infinite number of jobs in front of Station 1. The processing times of each job at Stations 1, 2, and 3 are

Table 2. Means and Standard Deviations of the Ratios of Estimation to the True Value of the Sample Sizes for the Illustrative Problem

n_0	KN		BIZ		GSP	
	mean	std	mean	std	mean	std
10	0.78	0.14	1.41	0.21	0.79	0.18
20	0.82	0.11	1.49	0.17	0.63	0.17
50	0.81	0.08	1.28	0.11	0.67	0.17

Table 3. Means and Standard Deviations of the True Sample Sizes for the Illustrative Problem ($\times 10^5$, Two Significant Figures)

n_0	KN		BIZ		GSP	
	mean	std	mean	std	mean	std
10	21	1.5	1.9	0.15	180	21
20	13	1.4	2.1	0.10	68	7.0
50	12	0.7	2.9	0.15	41	5.1

independently exponentially distributed with service rates r_1, r_2, and r_3, respectively. The overall objective is to maximize expected steady-state throughput by finding an optimal (integer-valued) allocation of buffer and service rates. The warm-up period consists of 2,000 released jobs starting from an empty system and the average throughput is then computed based on the time required for the following 10 jobs. In practice one would estimate the throughput on many more jobs than the following 10. We take the rather extreme choice of 10 jobs to increase the variability of the output for this problem, to better demonstrate the performance of our proposed methods in scenarios where the numbers of samples needed are high. For further details of this problem, see Henderson and Pasupathy (2014).

For our test, we set $R = B = 20$, so that $k = 3{,}249$, and we set $\alpha = 0.05$ and $\delta = 0.1$. It takes 0.008s to get one simulation sample on average, and there is no significant difference in simulation time between different systems. We test our estimation for all three procedures, KN, BIZ, and GSP, using simulation-of-the-simulation-process. To find out the impact of the initial sample size n_0 on the accuracy of the prediction and the computational effort of the prediction, we tried $n_0 = 10$, 20, and 50 for all cases. All other parameters of the procedures take their default values.

The ratios of estimates and true values are given in Table 2. All of the estimates are reasonably accurate, suggesting that the estimation procedure is practical. Further, for different values of n_0, the results show that there is no significant change in the accuracy of the prediction for this problem as n_0 varies. Table 3 gives statistics on the total sample size required by each of the procedures as n_0 varies. As n_0 increases, the total sample size increases only for BIZ, and not by a huge amount, though certainly by a nontrivial amount. For the other procedures, the sample size decreases, due to a more confident prediction of the system variances that is exploited in the procedures. (BIZ uses a heuristic to handle estimated variances.) This suggests that the recommendation for R&S algorithms of using n_0 on the order of 10 or 20 is only a rule of thumb, and other choices may be better. Taking n_0 larger is useful, though not essential, for our regression-based estimation procedure. The choice of $n_0 = 50$ seems reasonable in general.

The computational time spent running the R&S procedure and in predicting the necessary simulation budget is given in Table 4. For all procedures, the time spent on simulations dominates,

Table 4. Average Total Running Time for the R&S Procedure and for the Prediction of Running Time for the Illustrative Problem (Seconds, Two Significant Figures)

n_0	KN		BIZ		GSP	
	R&S	predicting	R&S	predicting	R&S	predicting
10	1,000	7.3	140	1.4	7,800	1.7
20	630	9.1	140	1.7	2,800	2.0
50	610	12	200	2.1	1,700	2.4

with the predictions requiring at most a few percent of the time spent running simulations. This model is very simple and getting one replication takes only milliseconds. For more complicated models the ratio between the length of the time for making the prediction and for running the R&S procedure would be even smaller. We conclude that our predictions introduce negligible overhead. As a side note, GSP required significantly more computation than KN or BIZ in this instance. That may be due to a poor choice of the tunable parameters for GSP, or a good choice for those of KN. We did not attempt to tune these parameters, since our focus is on predicting computational effort.

We conclude this section by providing a sense of how the predictions can be used to select the number of cores in a parallel implementation. In doing so, we use the mean sample size predictions in Table 3 as if they were the predictions of the simulation budget, and the running-time information in Table 4 as if it gave the predicted running time obtained by scaling the predicted simulation budgets by average computational time obtained in the prediction stage. Of course, this means we are using actual running time values rather than the predictions in our explanation, which we do simply, because it means we need not provide additional information on the predictions; the decision process is the same. The decision process is simply to choose the number of cores, c say, so that the total computation time divided by c is on the order of the desired wall-clock computation time. An additional check is that the total number of simulation replications needed is large enough relative to c that one would expect a well-implemented R&S procedure to be able to efficiently distribute the computation across the cores. If the number of replications per core is too small, e.g., less than a few hundred, then we should be cautious of the heuristic of dividing total computation by available cores, which assumes perfect utilization of cores.

The running-times in Table 4 for $n_0 = 50$ (our proposed choice of n_0) suggest that both KN and GSP will be time-consuming, requiring on the order of 600 and 1,700s of computation, respectively. If we want the R&S procedure to conclude within, say, 1 or 2min, then we would require on the order of 10 and 30 cores for the two procedures, respectively. The number of needed replications from Table 3 is so large that one expects a well-implemented R&S procedure to be able to efficiently use the cores, so we expect that the procedures would be very likely to finish within 2min. Similarly, if one wanted to finish within a few seconds, then we would instead need on the order of 600 and 2,000 cores for the two procedures, respectively. KN requires on the order of 1.2 million replications overall, again suggesting that the recommended number of cores (600) could be efficiently used. This would not be the case if we wanted to finish in a small fraction of a second, however, since then the number of replications per core gets so low that one expects synchronization issues and the like to play a role.

The quality of this approximation for wall-clock running times as a function of the number of cores depends on using reasonably efficient parallel implementations of R&S procedures. Several recent papers discuss such parallel implementations. Some state-of-the-art schemes and design principles are discussed in Hunter and Nelson (2017). In Luo et al. (2015), vector-filling (VF) procedures and asymptotic parallel selection procedures are proposed to efficiently exploit parallel computing environments. Ni et al. (2017) proposed the "good selection procedure," and they compared different parallel implementations on several criteria including synchronization and utilization of the cores. The Message-Passing Interface (MPI) implementation achieves greater than 50% utilization (the ratio between total simulation time to the summed-up wall-clock time over all cores) in all configurations. For implementations of parallel R&S procedures with reasonable utilization rates such as these, our approach gives sufficiently reliable predictions of wall-clock computation time, though of course our predictions may be very poor for poor implementations with very low utilization rates.

The running time for generating a single replication of this simple example is very small, so the order of the predictions above are modest. For more complex simulation models where the total computation required to complete all of the simulation replications for all systems, if run on a single processor, is on the order of days, the prediction tools outlined in this article become more important and relevant. This is especially the case in some cloud-computing settings where the setup time at the start of a computation can be on the order of minutes. (Our simple heuristic for estimating wall-clock running times ignores setup.) In such settings, for target wall-clock computing times that are on the order of hours or more, our procedure is highly relevant and worthwhile.

6 CONCLUSIONS

In this article, we explored the problem of predicting the simulation budget of R&S procedures, which is important when dealing with large-scale problems. We presented two approaches for estimating the total number of samples needed for a R&S procedure to terminate, both of which rely on the estimation of the problem configuration, which is not trivial. We develop a linear combination estimator that exhibits excellent performance in a realistic setting, and reasonable performance even in the slippage configuration. Experimental results for both synthetic test problems and a realistic problem together suggest that our approach is effective and sufficient for application.

REFERENCES

Jerry Banks. 1998. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice.* John Wiley & Sons.

Robert E. Bechhofer. 1954. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Ann. Math. Stat.* 25, 1 (1954), 16–39.

Patrick Billingsley. 1995. *Probability and Measure* (3rd ed.). Wiley, New York.

Jürgen Branke, Stephen E. Chick, and Christian Schmidt. 2007. Selecting a selection procedure. *Manage. Sci.* 53, 12 (2007), 1916–1932.

Leo Breiman. 1992. *Probability.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Chun-Hung Chen, Jianwu Lin, Enver Yücesan, and Stephen E. Chick. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynam. Syst.* 10, 3 (2000), 251–270.

Stephen E. Chick, Jürgen Branke, and Christian Schmidt. 2010. Sequential sampling to myopically maximize the expected value of information. *INFORMS J. Comput.* 22, 1 (2010), 71–80.

Stephen E. Chick and Koichiro Inoue. 2001a. New procedures to select the best simulated system using common random numbers. *Manage. Sci.* 47, 8 (2001), 1133–1149.

Stephen E. Chick and Koichiro Inoue. 2001b. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research* 49, 5 (2001), 732–743.

A. B. Dieker and Seong-Hee Kim. 2012. Selecting the best by comparing simulated systems in a group of three when variances are known and unequal. In *Proceedings of the Winter Simulation Conference.* IEEE, 490–496.

Edward J. Dudewicz and Siddhartha R. Dalal. 1975. Allocation of observations in ranking and selection with unequal variances. *Sankhyā: Indian J. Stat., Ser. B* (1975), 28–78.

Weiwei Fan, L. Jeff Hong, and Barry L. Nelson. 2016. Indifference-zone-free selection of the best. *Operat. Res.* 64, 6 (2016), 1499–1514.

Peter I. Frazier. 2014. A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. *Operat. Res.* 62, 4 (2014), 926–942.

Peter W. Glynn and Philip Heidelberger. 1990. Bias properties of budget constrained simulations. *Operat. Res.* 38 (1990), 801–814.

David Goldsman, Seong-Hee Kim, William S. Marshall, and Barry L. Nelson. 2002. Ranking and selection for steady-state simulation: Procedures and perspectives. *INFORMS J. Comput.* 14, 1 (2002), 2–19.

Shanti Swarup Gupta. 1956. *On a Decision Rule for a Problem in Ranking Means*. Ph.D. Dissertation. University of North Carolina at Chapel Hill.

Donghai He, Stephen E. Chick, and Chun-Hung Chen. 2007. Opportunity cost and OCBA selection procedures in ordinal optimization for a fixed number of alternative systems. *IEEE Trans. Syst., Man, Cybernet., Part C (Appl. Rev.)* 37, 5 (2007), 951–961.

Shane G. Henderson and Raghu Pasupathy. 2014. Simulation optimization library. Retrieved from http://www.simopt.org.

L. Jeff Hong. 2006. Fully sequential indifference-zone selection procedures with variance-dependent sampling. *Naval Res. Logist.* 53, 5 (2006), 464–476.

L. Jeff Hong, Jun Luo, and Ying Zhong. 2016. Speeding up pairwise comparisons for large scale ranking and selection. In *Proceedings of the Winter Simulation Conference*. IEEE Press, 749–757.

Susan R. Hunter and Barry L. Nelson. 2017. Parallel ranking and selection. In *Advances in Modeling and Simulation*. Springer, 249–275.

June Young Jung, Gary Blau, Joseph F. Pekny, Gintaras V. Reklaitis, and David Eversdyk. 2004. A simulation-based optimization approach to supply chain management under demand uncertainty. *Comput. Chem. Engineer.* 28, 10 (2004), 2087–2106.

Nabil Kahale. 2008. Analytic crossing probabilities for certain barriers by Brownian motion. *Ann. Appl. Probabil.* 18, 4 (2008), 1424–1440.

Seong-Hee Kim and A. B. Dieker. 2011. Selecting the best by comparing simulated systems in a group of three. In *Proceedings of the Winter Simulation Conference*. IEEE, 3992–4002.

Seong-Hee Kim and Barry L. Nelson. 2001. A fully sequential procedure for indifference-zone selection in simulation. *ACM Trans. Model. Comput. Simul.* 11, 3 (2001), 251–273.

Seong-Hee Kim and Barry L. Nelson. 2006. On the asymptotic validity of fully sequential selection procedures for steady-state simulation. *Operat. Res.* 54, 3 (2006), 475–488.

Jun Luo, L. Jeff Hong, Barry L. Nelson, and Yang Wu. 2015. Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. *Operat. Res.* 63, 5 (2015), 1177–1194.

Sijia Ma. 2016. RnS running time. Retrieved from https://github.com/sjasonma/RnSRunningTime.

Matthew S. Maxwell, Shane G. Henderson, and Huseyin Topaloglu. 2013. Tuning approximate dynamic programming policies for ambulance redeployment via direct search. *Stochast. Syst.* 3, 2 (2013), 322–361.

Barry L. Nelson and Souvik Banerjee. 2001. Selecting a good system: Procedures and inference. *IIE Trans.* 33, 3 (2001), 149–166.

Eric C. Ni, Dragos F. Ciocan, Shane G. Henderson, and Susan R. Hunter. 2017. Efficient ranking and selection in parallel computing environments. *Operat. Res.* 65, 3 (2017), 821–836.

Eric C. Ni, Shane G. Henderson, and Susan R. Hunter. 2014. A comparison of two parallel ranking and selection procedures. In *Proceedings of the Winter Simulation Conference*. IEEE Press, 3761–3772.

Yijie Peng, Edwin K. P. Chong, Chun-Hung Chen, and Michael C. Fu. 2018. Ranking and selection as stochastic control. *IEEE Trans. Automat. Control* 63, 8 (2018), 2359–2373.

Yijie Peng, Michael C. Fu, Peter W. Glynn, and Jianqiang Hu. 2017. On the asymptotic analysis of quantile sensitivity estimation by Monte Carlo simulation. In *Proceedings of the Winter Simulation Conference*. IEEE, 2336–2347.

Yosef Rinott. 1978. On two-stage selection procedures and related probability-inequalities. *Commun. Stat.-Theory Methods* 7, 8 (1978), 799–811.

Moshe Shaked and George Shanthikumar. 2007. *Stochastic Orders*. Springer Science & Business Media.

S. R. Srinivasa Varadhan. 2007. *Stochastic Processes*, vol. 16. American Mathematical Society.

Received January 2018; revised February 2019; accepted March 2019